A Study on Radial Basis Functions and its Applications

Dissertation submitted to
Visvesvaraya National Institute of Technology, Nagpur
in partial fulfillment of the requirements for the award of the degree

Master of Science

in

Mathematics

by

Ashita Verma

(MS23MTH012)

under the guidance of

Dr. G. Naga Raju

Department of Mathematics
Visvesvaraya National Institute of Technology
Nagpur - 440010, (India)

May, 2025

A Study on Radial Basis Functions and its Applications

Dissertation submitted to
Visvesvaraya National Institute of Technology, Nagpur
in partial fulfillment of the requirements for the award of the degree

Master of Science
in
Mathematics

by

Ashita Verma

(MS23MTH012)

under the guidance of

Dr. G. Naga Raju

Department of Mathematics
Visvesvaraya National Institute of Technology
Nagpur - 440010, (India)

May, 2025

© Visvesvaraya National Institute of Technology (VNIT) 2009

Department of Mathematics

Visvesvaraya National Institute of Technology, Nagpur

Declaration

I, Ashita Verma, hereby declare that this dissertation titled "A Study on Radial Basis Functions and its Applications" is carried out by me in the Department of Mathematics, Visvesvaraya National Institute of Technology, Nagpur. The work is original and has not been submitted earlier whole or in part for the award of any degree/diploma at this or any other Institution/University.

Name and Signature
Date:

Certificate

This is to certify that the dissertation titled "A Study on Radial Basis Functions and its Applications " is submitted by **Ashita Verma** in partial fulfillment of the requirements for the award of the degree of **Master of Science in Mathematics**, VNIT Nagpur. The work is comprehensive, complete and fit for final evaluation.

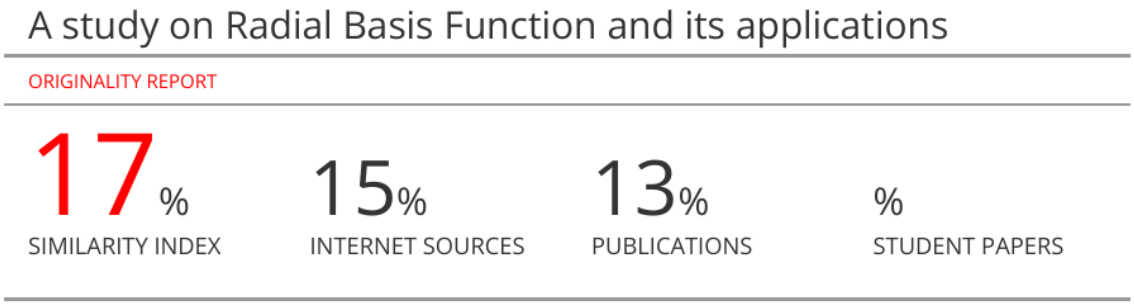
Dr. G. Naga Raju
Associate Professor, Mathematics,
VNIT, Nagpur

Head, Department of Mathematics
VNIT, Nagpur
Date:

VISVESVARAYA NATIONAL INSTITUTE OF TECHNOLOGY NAGPUR

Non-Plagiarism Certificate

Certified that M.Sc. Dissertation titled "**A Study on Radial Basis Functions and its Applications**" submitted by **Ashita Verma**, Enrollment No. **MS23MTH012**, ID No. **30339**, is original in nature and not copied from any literature material. Due to credit has been given to the original researcher wherever referred.



Signature
Name of the Student: **Ashita Verma**
Enrollment No: MS23MTH012

Dr. G. Naga Raju
Associate Professor
Department of Mathematics,
VNIT, Nagpur

Acknowledgement

First and foremost, I would like to express my profound gratitude to my supervisor, **Dr. G. Naga Raju**, for their invaluable guidance, insightful feedback, and unwavering support throughout the duration of this research. Their expertise and encouragement have been instrumental in shaping the direction and quality of this work. I would also like to extend my sincere thanks to the faculty members and staff of the **Department of Mathematics**, **VNIT Nagpur**, for their support and for providing an academic environment conducive to research and learning. My appreciation is further extended to my colleagues and peers, whose collaboration, discussions, and moral support have significantly contributed to the progress of this thesis. Lastly, I am deeply thankful to my family for their constant encouragement, patience, and understanding throughout the course of my academic journey. Their unwavering belief in me has been a source of strength and motivation.

Abstract

This dissertation offers an in-depth exploration of Radial Basis Functions (RBFs) as a mesh-free numerical method, emphasizing their use in interpolation, differentiation, and solving partial differential equations (PDEs). It begins by reviewing traditional interpolation methods, such as Lagrange and Newton's divided differences, and discusses their limitations—particularly when dealing with scattered or high-dimensional data. In contrast, RBF interpolation is introduced as a highly accurate and flexible technique that performs well over irregular domains.

The work presents the mathematical foundations of RBF interpolation in both one and two dimensions, accompanied by clear numerical examples. A comparative study with classical polynomial interpolation methods highlights the superior smoothness and accuracy of RBFs, especially for scattered data points. The dissertation further examines RBF-based numerical differentiation, showing it to be more effective than standard finite difference methods, particularly on non-uniform grids.

Additionally, the study extends use of RBFs to the numerical solution of time-dependent PDEs using strategies like the method of lines and the θ-method (weighted average method). Numerical experiments on equations such as the advection and heat equations demonstrate the robustness, stability, and precision of RBF-based approaches on both structured and unstructured node sets. Overall, the dissertation establishes RBFs as a highly capable and versatile tool in the field of computational mathematics.

Contents

1 Interpolation **2**
1.1 Introduction . . . 2
1.2 Lagrange Interpolation . . . 3
1.3 Newton Divided Difference Interpolation . . . 3
1.4 Radial Basis Function Interpolation . . . 4

2 Differentiation **10**
2.1 Numerical Differentiation . . . 10
2.1.1 Forward difference Approximation . . . 10
2.1.2 Backward difference Approximation . . . 10
2.1.3 Central difference Approximation . . . 11
2.2 RBF Differentiation . . . 11

3 Application in Partial Differential Equation **15**
3.1 Advection Equation . . . 15
3.1.1 Characteristic method . . . 15
3.1.2 Finite difference schemes . . . 16
3.2 Numerical solution of PDE using RBF approximation . . . 18

4 Conclusion **24**

Chapter 1

Interpolation

1.1 Introduction

Interpolation is a numerical method used in mathematics to estimate the values of a function at points located between known data values. When we do not have a analytical form for a function but only know its values at specific points, interpolation helps us build a function that passes through those points. This allow us to predict the value of the function at any position within the given range, which is especially useful in scientific experiments, engineering problems, and computer applications.
There are various interpolation methods available [**3**], and the choice of method depends on the nature of the data and the desired accuracy. Polynomial interpolation is among the most common approaches, where a single polynomial is constructed to pass exactly through all the given data points. Techniques such as Lagrange interpolation and Newton's divided difference method fall under this category and are widely used due to their straightforward implementation. Lagrange interpolation constructs a single polynomial that passes through all the given points, but it can suffer from oscillations, especially for higher-degree polynomial. Newton interpolation, on the other hand , builds the polynomial incrementally using divided difference making it more efficient and easier to update with new data. Both methods require structured or evenly spaced data to perform reliably. These methods are useful when the number of points are small, but for a large number of points, the resulting polynomial can become complex and lead to large errors or unwanted oscillations. This problem is known as Runge's phenomenon.
To address limitations of classical interpolation methods and to handle more complex, irregular data, especially in multiple dimensions, meshless methods have been developed. These methods are particularly powerful in cases where the structure of the data or the domain does not conform well to traditional mesh-based techniques.
This chapter gives a general overview of meshless method [**1**]. Since there are various application areas and this is a broad topic that is continually evolving, it will inevitably brief. In general, the meshless approach works quite well for appropriate problems and may be applied to a variety of applications, such as interpolation of scattered data and the solution of partial differential equation. It does not limit the shape of domain, therefore the domain's shape can be selected to make finding a solution for a certain application easier. In many real-world applications, we have scattered data that must be approximated as a function of several variables. A method known as the radial basis function has been devised to approximate multidimensional scattered data. There are no limitations on the placement of datapoints other than the requirement that they must be distinct from one

another, making it appropriate for this purpose since a distance function is computed between each datapoint and it's neighbours.

1.2 Lagrange Interpolation

Given the values of a function $f(x)$ at $n+1$ distinct points and ordered points $x_0, x_1, x_2, \cdots x_n$ where $x_0 < x_1 < x_2 < \cdots < x_n$, it is possible to construct a unique polynomial $P(x)$ of degree atmost n that passes exactly through these points. This polynomial satisfies the condition

$$P(x_i) = f(x_i), \quad i = 0, 1, 2, \cdots, n. \tag{1.1}$$

The highest degree of a polynomial that can satisfy the $n+1$ interpolation conditions given in equation (1.1) is atmost n. One common way to represent such a polynomial in the following general form:

$$P(x) = l_0(x)f(x_0) + l_1(x)f(x_1) + l_2(x)f(x_2) + \cdots + l_n(x)f(x_n) \tag{1.2}$$

where the functions $l_k(x)$, for $0 \leq k \leq n$, are polynomials of degree n. The expression given in equation (1.2) will satisfy the interpolation conditions stated in (1.1) if and only if

$$l_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i}, \quad k = 0, 1, 2, \cdots, n. \tag{1.3}$$

One can see that

$$l_k(x_i) = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{if } i \neq k \end{cases} \tag{1.4}$$

Each $l_k(x)$ given above is called lagrange's basis polynomial for the points $x_0, x_1, x_2, \cdots x_n$. The polynomial $l_k(x)$, which meets the conditions outlines in equation (1.4) , can be expressed in the following form

$$l_k(x) = \frac{(x - x_0)(x - x_1)(x - x_2) \cdots (x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0)(x_k - x_1)(x_k - x_2) \cdots (x_k - x_{k+1}) \cdots (x_k - x_n)} \tag{1.5}$$

equation (1.2) represents the Lagrange interpolation polynomial.
The error associated with approximating a function $f(x)$ using a Lagrange interpolating polynomial is given by

$$E_n(f; x) = \frac{f^{n+1}(\xi)}{(n+1)!}(x - x_0)(x - x_1) \cdots (x - x_n) \tag{1.6}$$

where ξ lies somewhere between smallest and largest of $(x_0, x_1, x_2, \cdots x_n, x)$. This shows that the error depends on the $(n+1)$th derivative of f and the node distribution.

1.3 Newton Divided Difference Interpolation

An interpolation polynomial that satisfies the conditions outlines in equation (1.1) can alternatively be expressed in the following form

$$P(x) = f[x_0] + (x - x_0)f[x_0, x_1] + \cdots + (x - x_0)(x - x_1) \cdots (x - x_{n-1})f[x_0, x_1, x_2, \cdots x_n] \tag{1.7}$$

The Newton basis functions used in this expression are defined as:

$$\begin{aligned}
\phi_0(x) &= 1 \\
\phi_1(x) &= (x - x_0) \\
\phi_2(x) &= (x - x_0)(x - x_1) \\
&\vdots \\
\phi_k(x) &= \prod_{j=0}^{k-1}(x - x_j), \quad \text{for } k = 1, 2, \ldots, n.
\end{aligned}$$

These basis functions, when multiplied by the corresponding divided differences, form the complete interpolation polynomial.

$$\begin{aligned}
f[x_0] &= f(x_0) \\
f[x_0, x_1] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} \\
f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \\
f[x_0, x_1, \cdots x_k] &= \frac{f[x_1, x_2, \cdots, x_k] - f[x_0, x_1, \cdots, x_{k-1}]}{x_k - x_0}
\end{aligned}$$

The terms represent the zeroth, first, second, and k-th order divided differences, respectively. The expression given in equation (1.7) is referred to as the Newton divided difference interpolating polynomial. The function $f(x)$ can be expressed as

$$f(x) = P(x) + R_{n+1}(x) \tag{1.8}$$

where $R_{n+1}(x)$ represents the remainder term. The polynomial $P(x)$ is of degree at most n and satisfies the interpolation constraint

$$f(x_k) = P(x_k), \quad k = 0, 1, 2, \cdots, n \tag{1.9}$$

it follows that the remainder $R_{n+1}(x)$ becomes zero at each note x_k for $k = 0, 1, 2, \cdots, n$. It is important to note that the interpolation polynomial defined by the conditions in equation (1.1) is unique.

1.4 Radial Basis Function Interpolation

A function $g : \mathbb{R}^s \to \mathbb{R}$ is called radial if its value depends solely on the magnitude(norm) of its input vector. Formally, such a function can be expressed as

$$g(x) = \varphi(\|x\|) = \varphi(r)$$

where $\varphi : [0, \infty) \to \mathbb{R}$ is a scalar function, and $r = \|x\|$ represents the Euclidean norm(length) of the vector x.
RBFs are often finite linear combinations of translates of radially symmetric basis functions, such as $\varphi(\|.\|)$, where $\|.\|$ stands for Euclidean norm. Due to Radial symmetry, rotations have no effect on the value of the function, instead it solely depends on the Euclidean distance of the argument of φ from the origin. Hardy (1971, 1990) introduced the multiquadratic RBF, which is currently frequently employed in multidimensional scattered interpolation, and thereby sparked the field of Radial basis function(RBFs).

Types of RBFs

Radial Basis Function	$\boldsymbol{\phi(r)}$	**parameters**
Gaussians	$\exp(-(er)^2)$	$e > 0$
Multiquadrics	$\sqrt{r^2 + e^2}$	$e > 0$
Inverse Multiquadrics	$\frac{1}{\sqrt{r^2 + e^2}}$	$e > 0$
Inverse Quadratics	$\frac{1}{r^2 + e^2}$	$e > 0$

Table 1.1: Common types of Radial Basis Functions

Interpolation

We assume that we can interpolate a function $f(x)$ represented by a scattered data using a radial basis function expansion-

$$\mathcal{P}_f(\mathbf{x}) = \sum_{k=1}^{N} c_k \varphi\left(\|\mathbf{x} - \mathbf{x}_N\|_2\right), \quad \mathbf{x} \in \mathbb{R}^s \tag{1.10}$$

In generally the components of x are represented as (x, y) in two dimensions and as (x, y, z) in three dimensions. Solving the following linear system yields the coefficient c_k:

$$\begin{bmatrix} \varphi(\|\mathbf{x}_1 - \mathbf{x}_1\|_2) & \varphi(\|\mathbf{x}_1 - \mathbf{x}_2\|_2) & \cdots & \varphi(\|\mathbf{x}_1 - \mathbf{x}_N\|_2) \\ \varphi(\|\mathbf{x}_2 - \mathbf{x}_1\|_2) & \varphi(\|\mathbf{x}_2 - \mathbf{x}_2\|_2) & \cdots & \varphi(\|\mathbf{x}_2 - \mathbf{x}_N\|_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(\|\mathbf{x}_N - \mathbf{x}_1\|_2) & \varphi(\|\mathbf{x}_N - \mathbf{x}_2\|_2) & \cdots & \varphi(\|\mathbf{x}_N - \mathbf{x}_N\|_2) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_N) \end{bmatrix} \tag{1.11}$$

which, as a matrix equation, we write $Ac = f$.
Each radial basis function is represented by the terms $\varphi(\|x_i - x_j\|_2) = \varphi(r_{i,j})$, where r stands for the Euclidean distance $(r_{i,j}) = \|x_i - x_j\|_2$.
Our ultimate goal is to solve -

$$\begin{bmatrix} \varphi(\|\mathbf{x}_1^e - \mathbf{x}_1^c\|_2) & \varphi(\|\mathbf{x}_1^e - \mathbf{x}_2^c\|_2) & \cdots & \varphi(\|\mathbf{x}_1^e - \mathbf{x}_N^c\|_2) \\ \varphi(\|\mathbf{x}_2^e - \mathbf{x}_1^c\|_2) & \varphi(\|\mathbf{x}_2^e - \mathbf{x}_2^c\|_2) & \cdots & \varphi(\|\mathbf{x}_2^e - \mathbf{x}_N^c\|_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(\|\mathbf{x}_M^e - \mathbf{x}_1^c\|_2) & \varphi(\|\mathbf{x}_M^e - \mathbf{x}_2^c\|_2) & \cdots & \varphi(\|\mathbf{x}_M^e - \mathbf{x}_N^c\|_2) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} \mathcal{P}_f(\mathbf{x}_1) \\ \mathcal{P}_f(\mathbf{x}_2) \\ \vdots \\ \mathcal{P}_f(\mathbf{x}_M) \end{bmatrix} \tag{1.12}$$

which, as a matrix equation, we write $Bc = \mathcal{P}_f$. In this equation we have imposed the condition -

$$\mathcal{P}_f(\mathbf{x}_i) = f(\mathbf{x}_i) \quad i = 1, 2 \cdots M \tag{1.13}$$

which represents the set of interpolated values we want to evaluate.
In order to find $\mathcal{P}_f$, We must first solve the linear system $Ac = f$ of eqn (1.11) for c, where the entries of A are provided by

$$A_{i,j} = \varphi(\|x_i - x_j\|_2) \quad i, j = 1, 2 \cdots N \tag{1.14}$$

In this case x_i and x_j stands for the collocation points and evaluation points respectively, these will generally be the same. Next, we use the previously computed vector c to solve the linear system $Bc = \mathcal{P}_f$. The entries of B are provided by

$$B_{i,j} = \varphi(\|x_i^e - x_j^c\|) \quad i = 1, 2 \cdots M, \quad j = 1, 2 \cdots N \tag{1.15}$$

The estimation or interpolation points in this case are collocation points x_i^e and the centres are denoted by x_j^c.

Interpolation Example: 1D

Consider a system of six scattered $1D$ data points:

$$x = [0.500 \quad 0.250 \quad 0.750 \quad 0.125 \quad 0.625 \quad 0.375]^T, \tag{1.16}$$

and the function to be interpolated is

$$f(x) = \exp(\sin(\pi x)). \tag{1.17}$$

Using the *multiquadratic* RBF,

$$\varphi(r) = \sqrt{(er)^2 + 1}, \tag{1.18}$$

We have employed a shape parameter value of $e = 1.2$, which was determined through trial and error.
(1.11) becomes

$$\begin{bmatrix} 1.0000 & 1.0440 & 1.0440 & 1.0965 & 1.0111 & 1.0111 \\ 1.0440 & 1.0000 & 1.1661 & 1.0111 & 1.0965 & 1.0111 \\ 1.0440 & 1.1661 & 1.0000 & 1.2500 & 1.0111 & 1.0965 \\ 1.0965 & 1.0111 & 1.2500 & 1.0000 & 1.1661 & 1.0440 \\ 1.0111 & 1.0965 & 1.0111 & 1.1661 & 1.0000 & 1.0440 \\ 1.0111 & 1.0111 & 1.0965 & 1.0440 & 1.0440 & 1.0000 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} 2.7182 \\ 2.0281 \\ 2.0281 \\ 1.4662 \\ 2.5190 \\ 2.5190 \end{bmatrix} \tag{1.19}$$

where $r_{i,j} = x_i^c - x_j^c$, $\varphi_{i,j} = \varphi(r_{i,j})$, $f_i = f(x_i^c)$ with $i, j = 1, 2, \cdots 6$. Solving the relationship $Ac = f$ of (1.19), we obtain,

$$c = [-4369.79 \ -737.37 \ -699.64 \ 11.21 \ 2876.8 \ 2918.8]^T \tag{1.20}$$

We can now estimate the value of a new set of data points by interpolating the scattered data. We determine the estimated function values at the following 20 points:

$$\begin{aligned} x^e = [&0.0000 \ 0.0526 \ 0.1052 \ 0.1578 \ 0.2105... \\ &0.2631 \ 0.3157 \ 0.3684 \ 0.4210 \ 0.4736... \\ &0.5263 \ 0.5789 \ 0.6315 \ 0.6842 \ 0.7368... \\ &0.7894 \ 0.8421 \ 0.8947 \ 0.9473 \ 1.0000]^T \end{aligned}$$

Then the estimated function values are given by:

$$\mathcal{P}_f(x_i^e) = Bc = \sum_{j=1}^{6} c_j \varphi(r_{i,j}^e), \quad i = [1, 2, \ldots, 20] \tag{1.21}$$

where B is a 20×6 matrix from equation (1.12) and $\varphi(r^e_{i,j}) = \varphi(x^e_i - x^c_j)$, c is the computed column vector of equation (1.20). Consequently, calculating the final result from above eqn, we get:

$$\begin{aligned}\mathcal{P}_f = [&1.0002\ 1.1790\ 1.3836\ 1.6095\ 1.8481\ldots\\ &2.0869\ 2.3098\ 2.4987\ 2.6363\ 2.7090\ldots\\ &2.7090\ 2.6363\ 2.4987\ 2.3097\ 2.0869\ldots\\ &1.8483\ 1.6101\ 1.3852\ 1.1820\ 1.0052]^T\end{aligned}$$

This compares with the exact values

$$\begin{aligned}f_{\text{exact}} = [&1.0000\ 1.1789\ 1.3836\ 1.6095\ 1.8482\ldots\\ &2.0869\ 2.3098\ 2.4987\ 2.6363\ 2.7090\ldots\\ &2.7090\ 2.6363\ 2.4987\ 2.3098\ 2.0869\ldots\\ &1.8482\ 1.6095\ 1.3836\ 1.1789\ 1.0000]^T\end{aligned}$$

which corresponds to a maximum absolute error of 0.005. Though we must keep in mind that the function $f(x)$ used in this example is a straightforward, well behaved smooth function, this is a pretty good result obtained from six scattered data points.

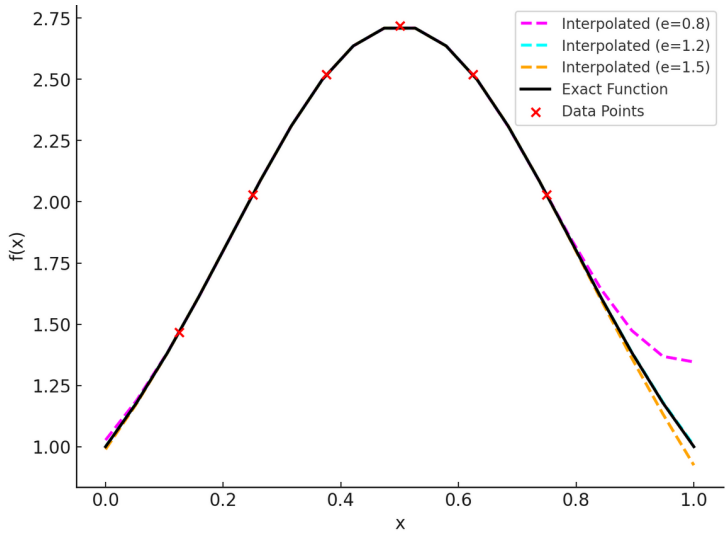


Figure 1.1: Radial basis function interpolation graph with different shape parameters

Comparison between RBF, Lagrange / Newton Interpolation

Consider a system of six scattered $1D$ data points (1.16) and the function to be interpolated is (1.17),

x_i	$f(x_i)$	Interpolating Polynomial $P(x)$
0.500	2.718282	$27.21x^5 - 31.54x^4 - 7.01x^3 + 10.53x^2 + 2.03x + 1.07$
0.250	2.028115	
0.750	2.028115	
0.125	1.466214	
0.375	2.519044	

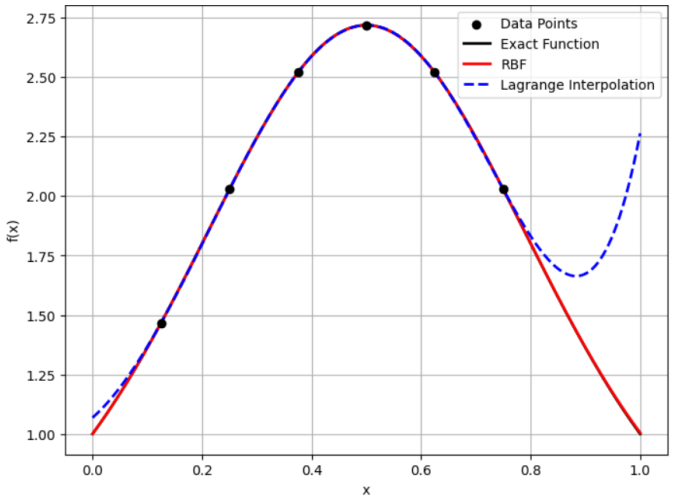


Figure 1.2: Comparison of RBF , Lagrange or Newton Interpolation

From the graph analysis of Lagrange / Newton and Radial Basis Functions(RBF) interpolation methods, it is evident that while all two techniques pass through the given data points, their behaviour between points differs significantly.

- Lagrange / Newton interpolating polynomial may cause noticeable oscillations, especially between widely spaced points.
- RBF interpolation polynomial is smoother and more flexible, especially effective for irregular or scattered data.

Interpolation example: 2D

Consider a system of four scattered (x, y) data points:

x	y
0.500	0.3333
0.250	0.6667
0.750	0.1111
0.125	0.4444

The function to be interpolated is:

$$f(x, y) = \sin(\pi x)\sin(\pi y) \tag{1.22}$$

Using the *multiquadratic* RBF:

$$\varphi(r) = \sqrt{(er)^2 + 1} \tag{1.23}$$

shape parameter value of $e = 20$, equation (1.11) becomes,

$$\begin{bmatrix} 1.0000 & 8.3911 & 6.7641 & 7.8859 \\ 8.3911 & 1.0000 & 14.9818 & 5.1964 \\ 5.1964 & 14.9818 & 1.0000 & 14.2019 \\ 7.8859 & 5.1964 & 14.2019 & 1.0000 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 0.8660 \\ 0.6123 \\ 0.2418 \\ 0.3768 \end{bmatrix} \tag{1.24}$$

where $r_{i,j} = \sqrt{(x_i^c - x_j^c)^2 - (y_i^c - y_j^c)^2}$, $\varphi_{i,j} = \varphi(r_{i,j})$, $f_{i,j} = f(x_i^c, y_j^c)$ with $i, j = 1, \cdots, 4$. Solving the relationship $Ac = f$ of eqn(1.11), we get,

$$c = [-0.0959,\ 0.0017,\ 0.0752,\ 0.0555]^T \tag{1.25}$$

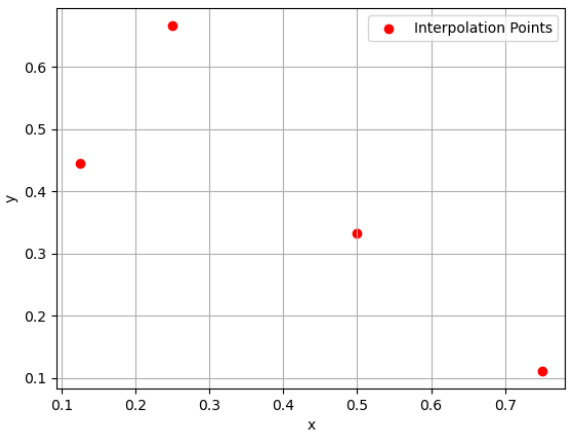
we can now estimate the value of a new datapoint by interpolating the scattered data. As an illustration, if we choose $x^e = 0.65, y^e = 0.65$, then the estimated function value is given by:

$$\mathcal{P}_f(x^e, y^e) = Bc = \sum_{i=1}^{4} c_i \varphi(r_i^e) \tag{1.26}$$

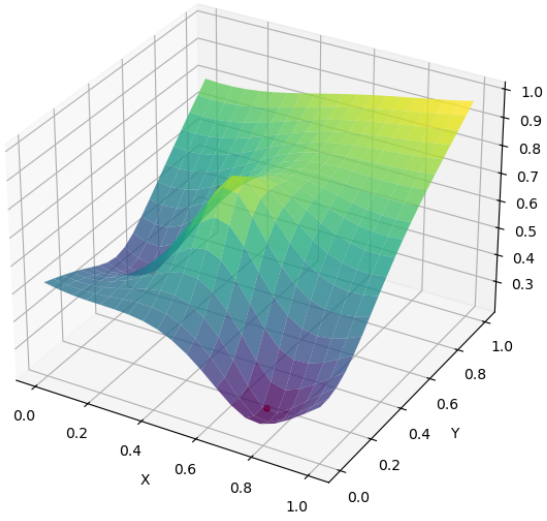
where $r_i^e = \sqrt{(x^e - x_i^c)^2 - (y^e - y_i^c)^2}$, c is the computed column vector of equation (1.25) and B is a 1×4 matrix from equation (1.12). Thus, calculating the final result for above eqn, we obtain,

$$\mathcal{P}_f(0.65, 0.65) = 0.7921 \tag{1.27}$$

This compares with the exact value $f_{exact} = 0.7938$, which corresponds to an error of 0.0017. This is quite incredible outcome, especially because it was obtained using only four scattered data points.



(a) Interpolation Points



(b) Interpolation using RBF in XY domain

Chapter 2

Differentiation

2.1 Numerical Differentiation

Numerical differentiation is employed to approximate the derivatives of a function when only discrete data points are available. Finite difference methods achieve this by constructing algebraic formulas based on Taylor series expansion about a point.

2.1.1 Forward difference Approximation

Let $f(x)$ be a smooth function defined on an interval containing x, and let $h > 0$ be a small step size. The forward difference formula for the first derivative is:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

using Taylor series expansion about x:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \cdots$$

subtracting $f(x)$ and dividing by h, we obtain

$$\frac{f(x+h) - f(x)}{h} = f'(x) + \frac{h}{2}f''(x) + \mathcal{O}(h^2)$$

Hence, the forward difference is first-order accurate with a local truncation error of $\mathcal{O}(h)$.

2.1.2 Backward difference Approximation

The Backward difference formula for the first derivative is given by

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

Expanding using Taylor series expansion:

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + \cdots$$

subtracting $f(x)$ and dividing by h, we obtain

$$\frac{f(x) - f(x-h)}{h} = f'(x) - \frac{h}{2}f''(x) + \mathcal{O}(h^2)$$

Thus, the backward difference is also first-order accurate with a local truncation error of $\mathcal{O}(h)$.

2.1.3 Central difference Approximation

The central difference formula provides a more accurate estimate of the first derivative:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

Taylor series expansion of $f(x+h)$ and $f(x-h)$ about x,

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \cdots$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + \cdots$$

subtracting we obtain,

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{2h^3}{6}f'''(x) + \cdots$$

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{h^2}{3}f'''(x) + \mathcal{O}(h^4)$$

This method is second-order accurate, with truncation error $\mathcal{O}(h^2)$.

2.2 RBF Differentiation

A straightforward application of the chain rule for differentiation can differentiate the RBF $\varphi(r)$, so for a two-dimensional domain in Cartesian coordinates, we obtain

$$\varphi(r) = \varphi(\|x\|) = \sqrt{x^2 + y^2}, \tag{2.1}$$

The First derivative with respect to x becomes;

$$\frac{\partial}{\partial x}\varphi(\|x\|) = \frac{x}{r}\frac{d}{dr}\varphi(r) \tag{2.2}$$

Similarly, the First derivative with respect to y;

$$\frac{\partial}{\partial y}\varphi(\|x\|) = \frac{y}{r}\frac{d}{dr}\varphi(r) \tag{2.3}$$

The Second derivative with respect to x;

$$\frac{\partial^2}{\partial x^2}\varphi(\|x\|) = \frac{x^2}{r^2}\frac{d^2}{dr^2}\varphi(r) + \frac{y^2}{r^3}\frac{d}{dr}\varphi(r) \tag{2.4}$$

Similarly, Second derivative with respect to y;

$$\frac{\partial^2}{\partial y^2}\varphi(\|x\|) = \frac{y^2}{r^2}\frac{d^2}{dr^2}\varphi(r) + \frac{x^2}{r^3}\frac{d}{dr}\varphi(r) \tag{2.5}$$

The Laplacian is simply given by;

$$\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)\varphi(\|x\|)=\frac{d^2}{dr^2}\varphi(r)+\frac{1}{r}\frac{d}{dr}\varphi(r) \tag{2.6}$$

In context of mixed partials, the second derivative becomes;

$$\frac{\partial^2}{\partial x\partial y}\phi(\|x\|)=\frac{xy}{r^2}\frac{d^2}{dr^2}\phi(r)-\frac{xy}{r^3}\frac{d}{dr}\phi(r) \tag{2.7}$$

Let us now consider the linear system $Ac=u$, where $u=[u_1,u_2,....u_N]$. This system can be differentiated once with respect to x to get $A_xc=u'$ and twice with repect to x to obtain $A_{xx}c=u''$. Since $D_x=A_xA^{-1}$ and $D_{xx}=A_{xx}A^{-1}$ are known as differentiation matrices, this result in $u'=D_xu$ and $u''=D_{xx}u$.

Differentiation example: 1D

The multiquadratic RBF will now be used to apply the previously discussed concepts for meshless differentiation. Consider the function $f(x)$ and the associated first derivative:

$$f(x)=x(x-1)^2, \tag{2.8}$$

We use the following six data points as the meshless centers:

$$x=[0.500\quad 0.250\quad 0.750\quad 0.125\quad 0.625\quad 0.375]^T, \tag{2.9}$$

For the MQ RBF $\varphi(r)=\sqrt{(er_c)^2+1}$, the corresponding A matrix with shape parameter $e=0.5$ is computed to be

$$A=\begin{bmatrix}1.00000 & 1.00778 & 1.00778 & 1.01742 & 1.00195 & 1.00195\\ 1.00778 & 1.00000 & 1.03077 & 1.00195 & 1.01742 & 1.00195\\ 1.00778 & 1.03077 & 1.00000 & 1.04769 & 1.00195 & 1.01742\\ 1.01742 & 1.00195 & 1.04769 & 1.00000 & 1.03077 & 1.00778\\ 1.00195 & 1.01742 & 1.00195 & 1.03077 & 1.00000 & 1.00778\\ 1.00195 & 1.00195 & 1.01742 & 1.00778 & 1.00778 & 1.00000\end{bmatrix} \tag{2.10}$$

where r_c is the absolute distance between centers. The corresponding A_x matrix for the first derivative of the RBF $\varphi_x=\dfrac{e^2r_x}{\sqrt{(er_c)^2+1}}$, is

$$A_x=\begin{bmatrix}0.00000 & 0.06202 & -0.06202 & 0.09214 & -0.03119 & 0.03119\\ -0.06202 & 0.00000 & -0.12127 & 0.03119 & -0.09214 & -0.03119\\ 0.06202 & 0.12127 & 0.00000 & 0.14914 & 0.03119 & 0.09214\\ -0.09214 & -0.03119 & -0.14914 & 0.00000 & -0.12127 & -0.06202\\ 0.03119 & 0.09214 & -0.03119 & 0.12127 & 0.00000 & 0.06202\\ -0.03119 & 0.03119 & -0.09214 & 0.06202 & -0.06202 & 0.00000\end{bmatrix} \tag{2.11}$$

where r_x represents the signed distance between centres.
The first derivative matrix is given by $D_x=A_xA^{-1}$ and the first derivative of a function is given by $\dfrac{df}{dx}=D_xf$, for f as given in (2.8) is obtained as

$$\frac{df}{dx} = [-0.24998,\ 0.18757,\ -0.31254,\ 0.54640,\ -0.32813,\ -0.07815]^T \tag{2.12}$$

we compare RBF derivative (2.12) with exact derivative,

$$\frac{df(x)}{dx} = (x-1)^2 + 2x(x-1) \tag{2.13}$$

which at grid points (2.9) is given by

$$\frac{df(x)}{dx} = [-0.25000,\ 0.18750,\ -0.31250,\ 0.54687,\ -0.32812,\ -0.07812]^T \tag{2.14}$$

The absolute errors between the numerical and exact derivatives are:

$$error = [0.00002,\ 0.00007,\ 0.00004,\ 0.00047,\ 0.00001,\ 0.00003]^T \tag{2.15}$$

The results are highly accurate and demonstrate the usefulness of meshless methods utilising the MQ RBF.

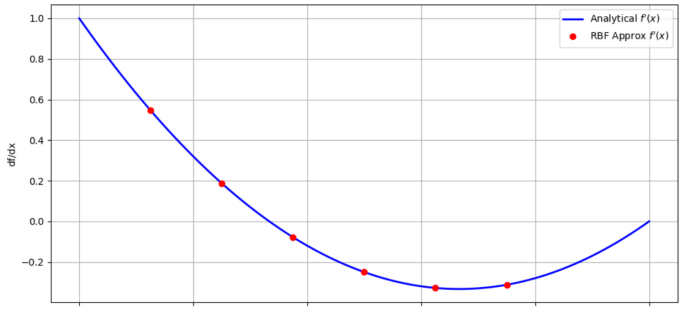


Figure 2.1: RBF Differentiation

Traditional differentiation rules

For $f(x)$ as in (2.8) at the points (2.9),

$$f'(x_i) = [-0.234375, 0.203125, -0.328125, 0.359375, -0.3125, -0.0625]^T \tag{2.16}$$

is obtained by

- **Forward difference** for first point, i.e. $i = 1$

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

- **Central difference** for interior points i.e. for $i = 2, 3, \cdots 5$

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}}$$

- **Backward difference** for last point i.e. $i = 6$

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

The absolute errors for this approximation (2.16) is as follows

$$error = [0.015625,\ 0.015625,\ 0.015625,\ 0.1875,\ 0.015625,\ 0.015625]^T \tag{2.17}$$

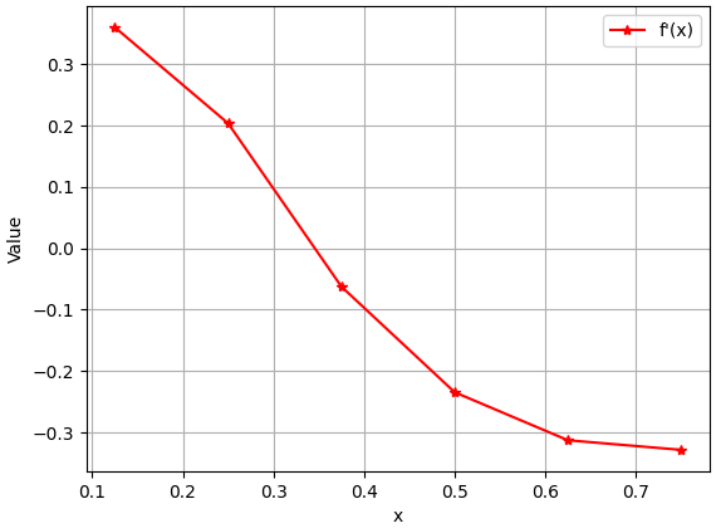


Figure 2.2: Plot of (2.16)

From the graphs (Figures (2.1) and (2.2)), it is evident that

- **Accuracy**: RBF gives more accurate derivatives, especially for scattered or irregular nodes.
- **Flexibility**: RBF works well on non-uniform grids, numerical methods prefer uniform spacing.
- **Error**: RBF shows smaller and more consistent errors than numerical methods.
- **Computation**: Numerical methods are simpler but less accurate, RBF needs matrix solving.
- **Graph**: RBF curve closely matches exact derivatives, numerical differentiation shows visible deviations.

Chapter 3

Application in Partial Differential Equation

3.1 Advection Equation

We analyze the one-dimensional linear transport equation[**2**]

$$u_t + a(x,t)u_x = 0 \tag{3.1}$$

In the most basic scenario, where $a(x,t) = a$. The above equation becomes

$$u_t + au_x = 0 \tag{3.2}$$

3.1.1 Characteristic method

To solve the ivp

$$u(x,0) = u_0(x) \quad \forall x \in \mathbb{R}.$$

The idea is that we have to reduce a PDE into an ODE along curves where the solution remains constant.

$$\begin{aligned}\frac{d}{dt}u(x(t),t) &= \frac{\partial u}{\partial t}(x(t),t) + \frac{d}{dt}x(t).\frac{\partial u}{\partial x}(x(t),t)\\ &= u_t + a(x,t)u_x\\ &= 0\end{aligned}$$

Since we assumed u is the solution of advection equation.
Implies that , if x(t) meets the ODE

$$\frac{d}{dt}x(t) = a(x,t)$$

$$x(0) = x_0$$

The solution x(t) of this equation is called *characteristics curve.*
Now if we take the velocity field $a(x,t) \equiv a$, where a is constant an explicit solution to the above mentioned characteristic equation is obtained as

$$x(t) = x_0 \ + \ at$$

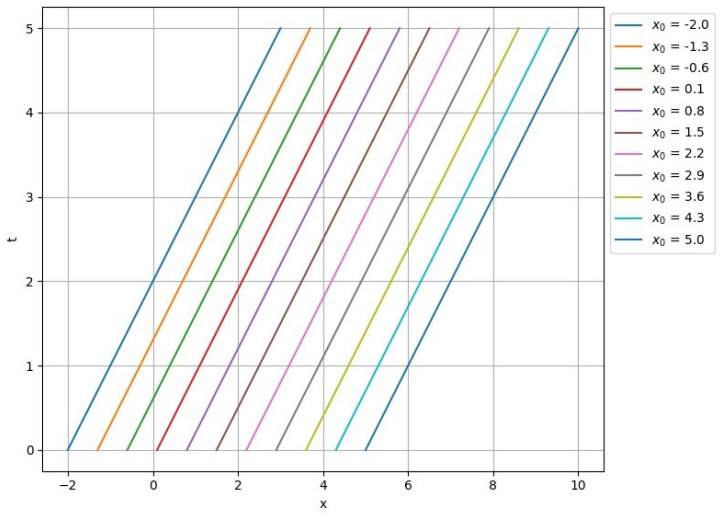


Figure 3.1: Characteristic curve x(t) for advection equation (3.1)

The characteristic equation is $x_0 = x - at$. Hence the solution of (3.2) is

$$u(x,t) = u_0(x - at); \quad \forall (x,t) \ \in \mathbb{R} \times \mathbb{R}_+ \tag{3.3}$$

The resultant formula (3.3) suggests that the speed a is used to convey the starting information.

Lemma 1. *If $u(x,t)$ is a continuous solution of $u_t + a(x,t)u_x = 0$, then $a \in C^1\ (\mathbb{R}, \mathbb{R}_+)$ breaks down to zero at infinite, that is, $\lim_{|x|\to\infty} u(x,t) = 0$ for all $x \in \mathbb{R}, t > 0$. Thus, u fulfills this energy bound.*

$$\int_{\mathbb{R}} u^2(x,t)dx \leq e^{||a||_{C^1}t} \int_{\mathbb{R}} u_0^2(x)dx; \quad \forall\ t > 0 \tag{3.4}$$

3.1.2 Finite difference schemes

When the field velocity is $a(x,t) = a$ is complex or nonlinear, solving the characteristic equation becomes difficult. Therefore, we apply numerical methods. Consider equation (3.2) with constant $a > 0$

Domain Discretization

We must truncate the spatial domain $\mathbb{R}$ to a finite domain $[x_L, x_R]$ because it is unbounded. As will be covered in more detail later, this truncation necessitates the introduction of appropriate boundary constraints.

For clarity, the domain $[x_L, x_R]$ is evenly divided with a grid size of Δx. $x_0 = x_L, x_N = x_R$, and $x_{j+1} - x_j = \Delta x$ for all j are the grid points x_j that are used for this. It is simple to consider an irregular discretization of data.

We select a terminal time T for the temporal discretization and split the time interval $[0, T]$ into M points. For $n = 0, 1, 2, ...M.$, this yields discrete time levels $t^n = n\Delta t$.

CFL Condition: Convergence of numerical method is possible only when, even in the limit as t and x approach zero, its numerical domain of reliance includes the PDE's genuine

domain of dependency..
It is important to realize that the CFL condition is just required for stability.

$$\nu = |a| \left(\frac{\Delta t}{\Delta x} \right) \leq 1$$

Upwind Scheme

For the transport equation (3.2), the central scheme violates the direction of information propagation. Therefore, we need to incorporate the proper direction of information dissemination and expect that it will stabilizes the scheme. To do this, the linear transport equation (3.2) must be approximated using one-sided differences rather than a central difference. If a>0 and the information propagates from left to right, then we can employ a backward difference in space to obtain the scheme.

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a \frac{u_j^n - u_{j-1}^n}{2\Delta x} = 0, \qquad j = 1, 2, ..., N-1 \tag{3.5}$$

and if a<0, we can use forward difference to obtain:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a \frac{u_{j+1}^n - u_j^n}{2\Delta x} = 0, \qquad j = 1, 2, ..., N-1 \tag{3.6}$$

Using the notation

$$a^+ = max\{a, 0\}, \ a^- = min\{a, 0\}, \ \mid a \mid = a^+ - a^-$$

(3.5) and (3.6) can be written together as

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a^+ \left(\frac{u_j^n - u_{j-1}^n}{\Delta x} \right) + a^- \left(\frac{u_{j+1}^n - u_j^n}{\Delta x} \right) = 0 \tag{3.7}$$

The above scheme considers the direction in which information spreads and is carried by the wind. This scheme is hence known as the upwind scheme.
Using the absolute value notion and a few basic algebraic operations, the upwind scheme (3.7), may be rewritten as

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = \frac{\mid a \mid}{2\Delta x} (u_{j+1}^n - 2u_j^n + u_{j-1}^n) \tag{3.8}$$

To keep things simple, we'll suppose that $a > 0$. Hence the upwind scheme (3.8) reduces to

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = \frac{a}{2\Delta x} (u_{j+1}^n - 2u_j^n + u_{j-1}^n) \tag{3.9}$$

It is comparable to the scheme (3.7) as well. We multiply both sides of the scheme (3.9) by u_j^n to get

$$u_j^n (u_j^{n+1} - u_j^n) = \frac{-a\Delta t}{2\Delta x} u_j^n (u_{j+1}^n - u_{j-1}^n) + \frac{a\Delta t}{2\Delta x} u_j^n (u_{j+1}^n - 2u_j^n + u_{j-1}^n) \tag{3.10}$$

Now we use elementary identity

$$y(x-y) = \frac{1}{2}x^2 - \frac{1}{2}y^2 - \frac{1}{2}(x-y)^2$$

a couple of times and rewrite (3.7) as

$$\frac{1}{2}(u_j^{n+1})^2 - \frac{1}{2}(u_j^n)^2 - \frac{1}{2}(u_j^{n+1} - u_j^n)^2 = -\frac{a\Delta t}{2\Delta x}(u_j^n u_{j+1}^n - u_j^n u_{j-1}^n)$$

$$+\frac{a\Delta t}{2\Delta x}\left[\frac{1}{2}(u_{j+1}^n)^2 - \frac{1}{2}(u_j^n)^2 - \frac{1}{2}(u_j^{n+1} - u_j^n)^2\right]$$

$$+\frac{a\Delta t}{2\Delta x}\left[\frac{1}{2}(u_{j-1}^n)^2 - \frac{1}{2}(u_j^n)^2 - \frac{1}{2}(u_{j-1}^n - u_j^n)^2\right]$$

Denoting

$$K_{j+1/2} = \frac{a}{2}\left(u_j^n u_{j+1}^n\right) - \frac{a}{4}[(u_{j+1}^n)^2 - u_j^n)^2]$$

and using

$$E^n = \frac{\Delta x}{2}\sum_j (u_j^n)^2$$

$$E^{n+1} = E^n + \frac{\Delta x}{2}\sum_j (u_j^{n+1} - u_j^n)^2 + \Delta t \sum_j (k_{j+1/2} - k_{j+1/2})$$

$$+\frac{a\Delta t}{4}\sum_j \left[(u_{j+1}^n - u_j^n)^2 + (u_{j-1}^n - u_j^n)^2\right]$$

After simplification we obtain,

$$E^{n+1} = E^n + \frac{\Delta x}{2}\sum_j (u_j^{n+1} - u_j^n)^2 + \frac{a\Delta t}{2}(u_j^n - u_{j-1}^n)^2 \tag{3.11}$$

Using the definition of the upwind scheme (3.7) in (3.11) yields

$$E^{n+1} = E^n + \left(\frac{a^2\Delta t^2}{2\Delta x} - \frac{a\Delta t}{2}\right)\sum_j \left(u_j^n - u_{j-1}^n\right)^2 \tag{3.12}$$

Hence, $E^{n+1} \leq E^n$,

Provided $\quad \dfrac{a^2\Delta t^2}{\Delta x} \leq a\Delta t \;\Rightarrow \left|\dfrac{a\Delta t}{\Delta x}\right| \leq 1$

3.2 Numerical solution of PDE using RBF approximation

We now describe a mesh-free approach that was first proposed by Kansa[Kan-90], who showed how RBFs can be combined with traditional numerical collocation methods to solve PDEs.
As can be seen from the preceding, with linear operator $\mathcal{L}$ with constant coefficients, we get,

$$\mathcal{L}u = A_{\mathcal{L}}c = A_{\mathcal{L}}A^{-1}u = D_{\mathcal{L}}u \tag{3.13}$$

where A and c are defined by eqn(1.11), $\mathcal{L}$ is made up of terms $\frac{d}{dx}, \frac{d^2}{dx^2}, ..,$ and so forth. Now consider PDE with Dirichlet boundary conditions:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \mathcal{L}u + f(x,t), \quad u = u(x,t), \quad x \in \Omega \subset \mathbb{R}^s, \quad s \in \{1,2,3\} \\ u(x,t) &= g(x,t), \quad x \in \partial\Omega, \\ u(x,0) &= u_0(x), \quad t = 0. \end{aligned} \tag{3.14}$$

Assumng $s = 1$, we now apply the concepts from section (2.2) to the system of equations (3.14). After discretising the problem throughout the spatial domain, we use meshless collocation, or $\mathcal{L}u = D_{\mathcal{L}}u$, to get,

$$\begin{aligned} \frac{du_i(t)}{dt} &= D_{\mathcal{L}}u_i(t) + f_i(t), \quad i = [1,2,...,N] \\ u_b(t) &= g_b(t), \\ u_i(0) &= u_{i,0} \end{aligned} \tag{3.15}$$

where the subscripts b and 0 denotes values of u_i on the boundary and at $t = 0$, respectively, and $u_i(t) = u(x = x_i, t)$ and $f_i(t) = f(x = x_i, t)$. Consequently, a PDE with the independent variable $u(x,t)$ has now been converted into a collection of ODEs with the independent variable $u_i(t)$, $i = [1,2,...,N]$. The analysis is equally applicable to a greater dimensions, even though we have assumed that $s = 1$.
Any common numerical integrator can be used to numerically intergrate equation (3.15) and produce a precise, stable solution.

Weighted Average Integration

We now consider the weighted average, or θ-method,

$$u^{n+1} = u^n + \Delta t \left[\theta\mathcal{L}u^{n+1} + (1-\theta)\mathcal{L}u^n\right] \tag{3.16}$$

which has been effectively applied to produce meshless solutions to a variety of PDE problems and has better stability properties than the explicit Euler method. It does, however, necessitate solving a matrix equation.
This scheme's application to eqn(3.14) yields,

$$(1 - \Delta t\theta\mathcal{L})u^{n+1} = (1 + \Delta t(1-\theta)\mathcal{L})u^n + \Delta t f^{n+1} \tag{3.17}$$

using RBF approximations and accounting for boundary conditions, we get,

$$\begin{aligned} \left[A^I - \Delta t\theta A^I_{\mathcal{L}}(x)\right] c^{n+1} &= \left(A^I + \Delta t(1-\theta)A^I_{\mathcal{L}}(x)\right) c^n + \Delta t f^{n+1}(x), \quad x \in \Omega^I, \\ A^B c^{n+1} &= g^{n+1}(x), \quad x \in \partial\Omega. \end{aligned} \tag{3.18}$$

where the boundary nodes on are indicated by superscript B, while the interior nodes within are indicated by superscript I. There are N_B boundary nodes and N_I interior nodes. As a result, $N = N_B + N_I$ is the total number of nodes. A^I and A^B are RBF matrices for interior and Boundary Nodes. The scheme can also be represented in matrix form, which is,

$$Gc^{n+1} = Hc^n + F^{n+1} \quad \Rightarrow \quad c^{n+1} = G^{-1}(Hc^n + F^{n+1}) \tag{3.19}$$

where the matrices are defined as

$$G = \begin{bmatrix} A^I - \Delta t\theta A^I_{\mathcal{L}} \\ A^B \end{bmatrix}, \quad H = \begin{bmatrix} A^I + \Delta t(1-\theta)A^I_{\mathcal{L}} \\ 0 \end{bmatrix}, \quad F^{n+1} = \begin{bmatrix} \Delta t f^{n+1} \\ g^{n+1} \end{bmatrix}$$

Consequently, we get the solution at t^{n+1} from

$$u^{n+1} = Ac^{n+1} \tag{3.20}$$

A less effective but comparable solution can be obtained by

$$\Rightarrow u^{n+1} = AG^{-1}HA^{-1}u^n + AG^{-1}F^{n+1} \tag{3.21}$$

Method of Lines

The method of lines is also applicable to the numerical solution of partial differential equations approximated through radial basis function (RBF) collocation. When this method is applied to the linear system

$$\frac{\partial u(x,t)}{\partial t} = \mathcal{L}u(x,t) + f(x,t),$$

while temporarily neglecting the boundary conditions, it results in the following discretized scheme:

$$u^n_t = \mathcal{L}u^n + f^n, \tag{3.22}$$

where $u = [u_1, u_2, ..., u_N]$ and $f = [f_1, f_2, ..., f_N]$ represents the discretization of continuous variables $u(x,t)$ and $f(x,t)$ respectively, superscript n denotes a variable assessed at the nth time step and the subscript t denotes differentiation with respect to time t.
We can now use the RBF approximation $D_{\mathcal{L}} = A_{\mathcal{L}}A^{-1}$ of eqn(3.13) to solve (3.22),

$$u^n_t = D_{\mathcal{L}}u^n + f^n \tag{3.23}$$

This resulting system of equations is subsequently integrated using a suitable standard ODE solver. Naturally, boundary conditions must be incorporated into the computation; these are typically enforced at the beginning of each time step. Similar to traditional non-meshless approaches, the effectiveness of a particular numerical integrator can vary depending on the specific problem context. Overall, the method of lines is straightforward to implement and generally produces accurate and reliable results.

Comparative analysis of the Upwind scheme and RBF method for Advection Equation

We examine the one-dimensional linear advection equation,

$$u_t + au_x = 0, \quad x \in [0,1], \quad t > 0$$

under given initial and Periodic boundary conditions

$$u(x,0) = \sin(2\pi x)$$

The exact solution is

$$u(x,t) = \sin 2\pi(x - at)$$

Using the Method of Lines, we get,

$$u_t^n = D_{\mathcal{L}} u^n \tag{3.24}$$

We have used the fourth-order Runge–Kutta (RK4) ODE integrator to solve Equation (3.24), employing the multiquadric (MQ) radial basis function defined as

$$\varphi(r) = \sqrt{(er)^2 + 1}.$$

The shape parameter $e = 1.80$ was selected through trial and error. The solution is plotted at the final time level $T = 1.001$, with parameters $a = 1$, time step $\Delta t = 0.007$, and the spatial derivative approximated using the RBF-based matrix $D_x = A_x A^{-1}$, where D_x approximates the first derivative.

First-Order Upwind Method Analysis

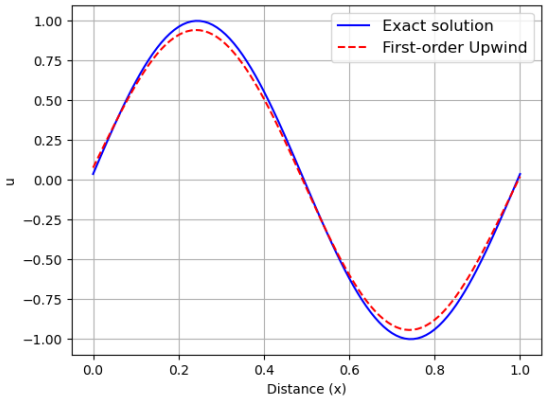


Figure 3.2: Numerical and exact solution for upwind method with equally spaced points

- **Dissipation:** Noticeable amplitude decay; peak values are lower than exact solution by about 5–10%.

- **Dispersion:** Phase lag present; wave travels slower than expected with increasing phase error over time.

- **Accuracy:** Moderate accuracy with visible differences from exact; method stable but overly dissipative due to artificial viscosity.

RBF + RK4 Method Analysis

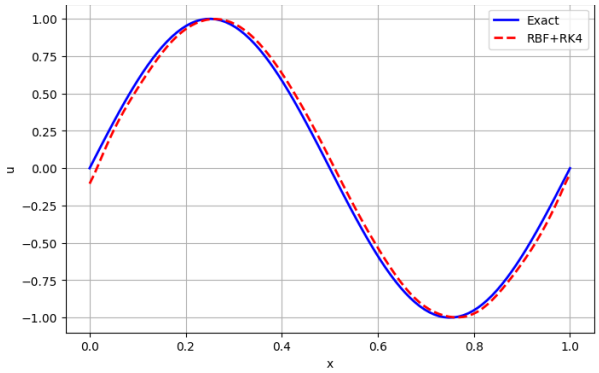


Figure 3.3: Numerical and exact solution for RBF method with equally spaced points

- **Accuracy:** Excellent agreement with exact solution; amplitude and phase preserved with high fidelity.
- **Numerical Errors:** Minimal dissipation and dispersion; amplitude maintained and phase accurately represented.
- **Spectral Accuracy:** Exhibits spectral-like properties; RK4 time integration ensures high temporal accuracy.

Example 1. *We aim to numerically solve the* $1D$ *heat equation*

$$u_t = u_{xx}, \quad x \in [0,1], \quad t > 0 \tag{3.25}$$

under given initial and boundary conditions

$$u(x,0) = \sin \pi x,$$
$$u(0,t) = u(1,t) = 0$$

The exact solution is

$$u(x,t) = \exp(-\pi^2 t) \sin \pi x$$

The weighted average(θ-method) for time discretization is:

$$u^{n+1} = u^n + \Delta t \left[\theta D_{xx} u^{n+1} + (1-\theta) D_{xx} u^n\right] \tag{3.26}$$

For the multiquadric (MQ) radial basis function $\varphi(r) = \sqrt{(er)^2 + 1}$, the shape parameter was chosen as $e = 34$, determined through trial and error. The time discretization uses $\theta = 0.5$ and time step $\Delta t = 0.001$. The spatial second derivative is approximated using the RBF-based matrix $D_{xx} = A_{xx}A^{-1}$, where D_{xx} approximates the second derivative operator. The solution is evaluated at the final time level $(T) = 0.1$.

In this example, a detailed computation of D_{xx}, and intermediate numerical steps, and final solution values are not present. The focus is on describing the numerical method used and illustrating the results through graphical comparison with eact solution.

Node Distribution Cases

To evaluate the flexibility and performance of the RBF-based numerical scheme, two types of spatial discretizations are considered:

- **Equally Spaced Points:** The spatial domain $[0, 1]$ is divided into uniform grid points. This structured configuration provides a symmetric and stable setup for derivative approximation.
- **Scattered Points:** A non-uniform node distribution is generated using a quasi-random (Halton) sequence. This case demonstrates the mesh-free capability of the RBF method and its adaptability to unstructured grids.

By examining both equally spaced and scattered node distributions, the objective is to demonstrate the adaptability of RBF-based numerical schemes in handling both structured and unstructured spatial data. This comparison provides insight into the method's accuracy, stability, and overall effectiveness under varying spatial configurations.

Graphical Results

The following plots show the numerical solution compared with the exact solution for both cases. These visual comparisons highlight the accuracy of the method on uniform as well as scattered nodes.

Case 1: Equally Spaced Points

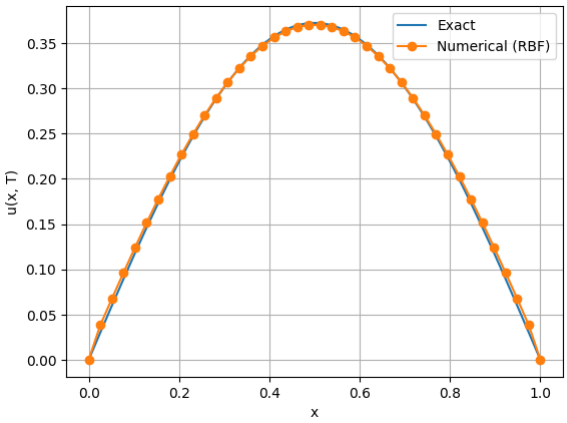


Figure 3.4: Heat equation solved with equally spaced points

- Numerical solution closely matches the exact solution curve.
- High accuracy is observed throughout the domain.
- Uniform spacing contributes to a stable and symmetric approximation.
- Demonstrates the strong performance of the RBF-based weighted average method on structured grids.

Case 2: Scattered Points

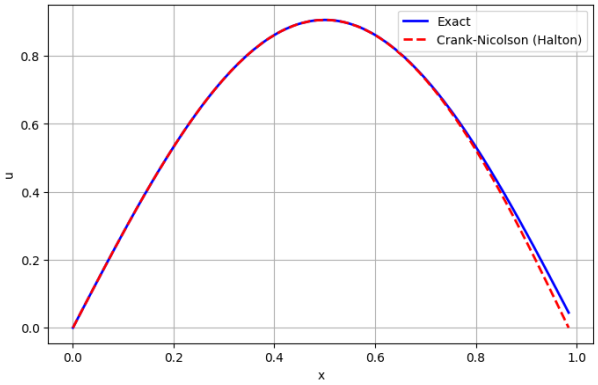


Figure 3.5: Heat equation solved with scattered points

- The overall solution profile is captured well.
- Slight deviations are noticeable near the domain boundaries.
- Scattered nodes lead to less uniform accuracy compared to the structured case.
- Highlights the flexibility of RBFs in solving PDEs without relying on a structured mesh.

Chapter 4

Conclusion

This work presented Radial Basis Functions as a flexible and accurate approach for interpolation, differentiation, and solving partial differential equations. Unlike traditional techniques, RBF methods perform well on scattered and irregular data without requiring a mesh. The results highlight their stability, efficiency, and potential for solving complex numerical problems.

Future Work

This study can be extended by optimizing shape parameters and applying RBF methods to nonlinear and higher-dimensional problems. Their use in complex domains and integration with machine learning techniques also offers promising directions for future research.

Bibliography

[1] Griffiths, Graham W.*Numerical Analysis Using R, Solutions to ODEs and PDEs*, Cambridge University Press, 2016.

[2] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, SIAM, 2007.

[3] Jain, Mahinder Kumar, Satteluri RK Iyengar, and Rajendra K. Jain. *Numerical methods: problems and solutions*. New Age International, 2007.